The fourth internal threat to validity

In Section 3.4, we stated that, given the same cut-off, *Precision* will be coincident with *Recall* as well as *F1*. Such a statement only fits in with unsupervised approaches such as our ClassRank, but not suitable for supervised approaches. It is mainly because the *data-splitting* operation for constructing training data set and testing data set will affect the calculation of these metrics.

Generally*, TP, FP, FN, TN, Precision*, *Recall*, and *F1* are defined as follows.

- True Positive (*TP*) refers to the number of true important classes that are also predicted by a specific approach as important ones.
- False Positive (*FP*) refers to the number of true unimportant classes that are predicted by a specific approach as important ones.
- False Negative (*FN*) refers to the number of true important classes that are predicted by a specific approach as unimportant ones.
- True Negative (*TN*) refers to the number of true unimportant classes that are also predicted by a specific approach as unimportant ones.

$$precision = \frac{TP}{TP+FP}$$

Mathematically, *Precision* represents the ratio of *TP* to the sum of *TP* and *FP,* and it is usually used to measure an approach's ability to correctly predict the positives out of all the positive prediction the approach made.

$$recall = \frac{TP}{TP+FN}$$

Mathematically, *Recall* represents the ratio of *TP* to the sum of *TP* and *FN*, and it is usually used to measure an approach's ability to correctly predict the positives out of actual positives.

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

Mathematically, *F1* is the harmonic mean of *Precision* and *Recall*, and it is useful when one needs to take both *Precision* and *Recall* into account.

Note that, for ***unsupervised*** approaches, we use a threshold *k* to further filter out unimportant classes, that is, the top-|*C*(*S*)|∗*k*% (|*C*(*S*)| is the number of classes in the system) ranked classes are the *predicted* important classes. Thus, (*TP*+*FP*) actually equals to |*C*(*S*)|∗*k*%, and (*TP* + *FN*) equals to the number of true important classes in the target system (i.e., the number of important classes in the *gold set*). Thus, in a specific subject system, (*TP*+*FP*) and (*TP*+*FN*) are all fixed under a specific *k* setting. Obviously, when comparing two approaches on a subject system under a specific *k* setting, their differences in *Precision*, *Recall*, and *F1* are mainly determined by *TP* (i.e., the numerator in Equations (6) and (7) in the revised manuscript). Thus, an approach with a better *Precision* also has a better *Recall* and *F1*; it seems that there is no need to compute *Precision*, *Recall*, and *F1* at the same time. Note that such an observation only fits in with *unsupervised* approaches.

However, for ***supervised*** approaches, the situation is different. In a supervised model, the data set splits into two pieces: one for training and the other for testing. Thus, there are some true important classes used for training; these important classes cannot be used for testing. (*TP*+*FP*) is the *predicted* important classes; (*TP*+*FN*) equals to the number of true important classes in the *testing* data set. In supervised approaches, the threshold *k* is applied to the testing data set, that is, the top-(*cte*∗*k*%) (*cte* is the number of classes in the testing data set) ranked classes are the *predicted* important classes.

Thus, for a specific data splitting and a specific *k* setting, (*TP*+*FP*) and (*TP*+*FN*) are all fixed. However, for two different rounds of data splitting, (*TP*+*FP*) can be fixed if we keep the ratio of *ctr* (number of classes in the *training* data set) to *cte* (the number of classes in the *testing* data set) the same; but (*TP*+*FN*) cannot be fixed since the distribution of true important classes over a training data set and testing data set may change greatly.

We use the following two examples to reproduce a ***puzzled phenomenon***.

Example 1: the comparison between two supervised approaches

Suppose that there is a system composed of 100 classes, among which 10 classes are important; two supervised approaches: Methods *A* and *B*.

Method *A***:** In one data splitting, the training data set contains 80 classes (where 7 classes are important), and the testing data set contains 20 classes (where 3 classes are important). If *k*=10, then top-2 (i.e., 20*10%=2) ranked classes are predicted as important classes (suppose only one class is true important), then we can obtain that *Precision*=1/2 and *recall*=1/3.

Method *B***:** In one data splitting, the training data set still contains 80 classes (where 2 classes are important), and the testing data set contains 20 classes (8 classes are important). If *k*=10, then top-2 (i.e., 20*10%=2) ranked classes are predicted as important classes (suppose the two classes are true important), then we can obtain that *Precision*=2/2 and *recall*=2/8=1/4.

If we compare the two methods with regard to *Precision*, then we may conclude that methods *B* is superior to *A* since 2/2>1/2. However, if we compare the two approaches with regard to *Recall*, we may conclude that methods *A* performs better than *B* since 1/3>1/4. These are two diametrically opposed conclusions.

This example reproduces the ***puzzled phenomenon*** between two supervised approaches.

Example 2: the comparison between one unsupervised approach and one supervised approach

Suppose that there is a system composed of 100 classes, among which 10 classes are important; one supervised approach Methods *A*, and one unsupervised approach Method *B*.

Method A: In one data splitting, the training data set contains 80 classes (where 7 classes are important), and the testing data set contains 20 classes (where 3 classes are important). If *k*=10, then top-2 (i.e., 20*10%=2) ranked classes are predicted as important classes (suppose only one class is true important), then we can obtain that *Precision*=1/2 and *recall*=1/3.

Method B: For unsupervised approaches, there is no training process, and they use the whole data set. If *k*=10, then top-10 (i.e., 100*10%=10) ranked classes are predicted as important classes (suppose four classes are true important), then we can obtain that *Precision*=4/10 and *recall*=4/10.

If we compare the two methods with regard to *Precision*, then we may conclude that methods *A* is superior to *B* since 1/2>4/10. However, if we compare the two approaches with regard to *Recall*, we may conclude that methods *B* performs better than *A* since 4/10>1/3. These are two diametrically opposed conclusions.

This example reproduces the ***puzzled phenomenon*** between one unsupervised approach and one supervised approach.

In summary, we believed that such a ***puzzled phenomenon*** mainly results from the data-splitting operation for constructing training data set and testing data set; the data-splitting operation will affect the calculation of these metrics. The deeper reason may be that there exists the *class-imbalance* problem in our data set (cf. Table 6 on Page 11 in the revised manuscript), that is, there are much more non-important classes (majority) than the important ones (minority); the supervised approaches proposed in the literature did not take any effective techniques to handle the *class-imbalance* problem. If there is no class-imbalance problem (i.e., the ratio of *important classes* to *un-important classes* is close to 0.5), then the distribution of true important classes in the training data set and testing data set might be more even; the ***puzzled phenomenon*** might be mitigated partially.

Note that the ANN classifier used in Static, VSM, Static_FS, VSM_FS, and VSM is the same as that used in the work of [1] and [2]. In fact, their replication packages contain the source code of their implementations. We think that the authors should improve their approach by introducing some effective remedies to handle the *class-imbalance* problem.

According to the discussion in the work of Zhang et al. [3], and Menzies et al. [4], it seems that, for classification problem with *class-imbalance* problem, *Recall* is a generally accepted metric, but *Precision* is not. Moreover, for the problem of ``*important class to document*" identification, we want to find the important classes in the gold set as much as possible. Thus, we prefer to use *Recall* to compare different approaches.

Note that, when computing *Recall*=*TP*/(*TP*+*FN*), the denominator (*TP*+*FN*) for supervised models is different from that for unsupervised models; generally, the latter is larger than the former. In this sense, comparing a supervised model with an unsupervised model might contain some unfairness, and thus the obtained conclusion may be not always true. We have discussed this threat in Section 5.1 on Page 31 (cf. the fourth internal threat to validity), and we have also highlighted this issue in the first paragraph of Section 3.5.1 on Page 14.

If the *class-imbalance* problem can be resolved, and an effective technique to handle class-imbalance problem can be applied, such an unfair comparison might be mitigated partially, and thus the comparison can be much more reasonable and fair. Though the conclusion might not be conclusive, it indeed can shed some light on the relative performance of supervised and unsupervised approaches. As such, we put such a comparison in our manuscript. We have highlighted the above issue many times in our manuscript, with the aim to remind the reader of the issue.

[1] P. W. McBurney, S. Jiang, M. Kessentini, N. A. Kraft, A. Armaly, M. W. Mkaouer, and C. McMillan, “Towards prioritizing documentation effort,” IEEE Trans. Software Eng., vol. 44, no. 9, pp. 897–913, 2018. [Online]. Available: https://doi.org/10.1109/TSE.2017.2716950

[2] S. Liu, Z. Guo, Y. Li, H. Lu, L. Chen, L. Xu, Y. Zhou, and B. Xu, “Prioritizing documentation effort: Can we do better?” CoRR, vol. abs/2006.10892, 2020. [Online]. Available:
https://arxiv.org/abs/2006.10892

[3] H. Zhang and X. Zhang, “Comments on ”data mining static code attributes to learn defect predictors”,” IEEE Trans. Software Eng., vol. 33, no. 9, pp. 635–637, 2007. [Online]. Available: https://doi.org/10.1109/TSE.2007.70706

[4] T. Menzies, A. Dekhtyar, J. S. D. Stefano, and J. Greenwald, “Problems with precision: A response to ”comments on ’data mining static code attributes to learn defect predictors’”,” IEEE Trans. Software Eng., vol. 33, no. 9, pp. 637–640, 2007. [Online]. Available: https://doi.org/10.1109/TSE.2007.70721